

SPECIFICATION AMENDMENTS

1. On page 1, lines 22-27, please amend as follows:

"The present U.S. Utility Patent Application is also a continuation-in-part of U.S. Utility Patent Application Serial No. 10/723,574, entitled "LDPC (Low Density Parity Check) coded modulation hybrid decoding," (Attorney Docket No. BP3134), filed November 26, 2003 (11/26/2003), ~~pending~~ now U.S. Patent 7,185,270 B2, issued on February 27, 2007 (02/27/2007), which is hereby incorporated herein by reference in its entirety and made part of the present U.S. Utility Patent Application for all purposes."

2. On page 33, lines 8-15, please amend as follows:

"An irregular LDPC code may also be described using a bipartite graph. However, the degree of each set of nodes within an irregular LDPC code may be chosen according to some distribution. Therefore, for two different variable nodes, v_{i_1} and v_{i_2} , of an irregular LDPC code, $|E_v(i_1)|$ may not equal to $|E_v(i_2)|$. This relationship may also hold true for two check nodes. The concept of irregular LDPC codes was originally introduced within M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical Loss-Resilient Codes," *Proc. 29th Symp. on Theory of Computing*, 1997, pp. 150-159 ~~M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman and V. Stemann, "Practical loss-resilient codes," *IEEE Trans. Inform. Theory*, Vol. 47, pp. 569-584, Feb. 2001.~~

3. On page ⁵⁴~~51~~, line 22 until page 55, 28, please amend as follows: SB 03-01-08

"When performing the update of the edge messages within the check node update functional block, the check node update functional block updates the check ~~bit~~ edge messages using the updated bit edge messages passed by the last iteration. During a first decoding iteration, this may include using the initialized values of the bit edge message. However, during the iterative decoding processing, the check node update functional block passed the updated edge messages to the symbol sequence estimate and symbol node update functional block.